Dr. D. Y. Patil Educational Federation's
Dr. D. Y. PATIL COLLEGE OF ENGINEERING & INNOVATION
Department of Artificial Intelligence and Data Science
Academic Year 2025-26

Academic Year: 2025-26

LABORATORY MANUAL

Name of the Student:		
Class: BE	**Division: B**	**Roll No.:**
Subject: Computer Laboratory I **(2019 Course) [417525]**		**Exam Seat No.:**
Department of Artificial Intelligence and Data Science		

Program Outcomes (PO's):

POs are statements that describe what students are expected to know and be able to do upon graduating from the program. These relate to the skills, knowledge, analytical ability attitude and behavior that students acquire through the program.

- **PO1: Engineering Knowledge:**
  Graduates will be able to apply the Knowledge of the mathematics, science and engineering fundamentals for the solution of engineering problems related to IT.
- **PO2: Problem Analysis:**
  Graduates will be able to carry out identification and formulation of the problem statement by requirement engineering and literature survey.
- **PO3: Design/Development of Solutions:**
  Graduates will be able to design a system, its components and/or processes to meet the required needs with consideration for public safety and social considerations.
- **PO4: Conduct Investigations of Complex Problems:**
  Graduates will be able to investigate the problems, categorize the problem according to their complexity using modern computational concepts and tools.
- **PO5: Modern Tool Usage:**
  Graduates will be able to use the techniques, skills, modern IT engineering tools necessary for engineering practice.
- **PO6: The Engineer and Society:**
  Graduates will be able to apply reasoning and knowledge to assess global and societal issues
- **PO7: Environment and Sustainability:**
  Graduates will be able to recognize the implications of engineering IT solution with respect to society and environment.
- **PO8: Ethics:**
  Graduates will be able to understand the professional and ethical responsibility.

- **PO9: Individual and Team Work:**

  Graduates will be able to function effectively as an individual member, team member or leader in multi -disciplinary teams.

- **PO10: Communication:**

  Graduates will be able to communicate effectively and make effective documentations and presentations.

- **PO11: Project Management and Finance:**

  Graduates will be able to apply and demonstrateengineering and management principles in project management as a member or leader.

- **PO12: Life-long Learning:**

  Graduates will be able to recognize the need for continuous learningand to engage in life-long learning.

Course Objectives and Course Outcomes (COs)

Course Objectives:

- Apply regression, classification and clustering algorithms for creation of ML models
- Introduce and integrate models in the form of advanced ensembles.
- Conceptualized representation of Data objects.
- Create associations between different data objects, and the rules.
- Organized data description, data semantics, and consistency constraints of data

Course Outcomes:

On completion of the course, students will be able to–

CO1: Implement regression, classification and clustering models

CO2: Integrate multiple machine learning algorithms in the form of ensemble learning.

CO3: Apply reinforcement learning and its algorithms for real world applications.

CO4: Analyze the characteristics, requirements of data and select an appropriate data model.

CO5: Apply data analysis and visualization techniques in the field of exploratory data science

CO6: Evaluate time series data.

Dr. D. Y. Patil Educational Federation's

Dr. D. Y. PATIL COLLEGE OF ENGINEERING & INNOVATION

Department of Artificial Intelligence and Data Science

Academic Year 2024-25

CERTIFICATE

This is to certify that Mr. /Ms. ____

of Class BE - AI-DS, Roll No.____Examination Seat No. __________________________

has completed all the practical work in the Computer Laboratory - I [417525] satisfactorily, as prescribed by Savitribai Phule Pune University, Pune in the academic year 2024-25 (Term-I).

Place:

Date:

Course In-charge **Department of AI-DS**	**HOD** **Department of AI-DS**	**Principal** **DYPCOEI, Varale**

INDEX

Department of Artificial Intelligence and Data Science **Class:** B.E.

Sr. No.	Name of the Experiment	Date of Conduction	Date of Checking	Page No.	Sign	Remark
1	**Assignment on PCA:** To use PCA Algorithm for dimensionality reduction. You have a dataset that includes measurements for different variables on wine (alcohol, ash, magnesium, and so on). Apply PCA algorithm & transform this data so that most variations in the measurements of the variables are captured by a small number of principal components so that it is easier to distinguish between red and white wine by inspecting these principal components.					
2	**Assignment on Predict the Price of the Uber Ride:** Predict the price of the Uber ride from a given pickup point to the agreed drop-off location. Perform following tasks: 1. Pre-process the dataset. 2. Identify outliers. 3. Check the correlation. 4. Implement linear regression and ridge, Lasso regression models. 5. Evaluate the models and compare their respective scores like R2, RMSE, etc.					

3	**Assignment on SVM:** Implementation of Support Vector Machines (SVM) for classifying images of handwritten digits into their respective numerical classes (0 to 9).					
4	**Assignment on K-Means Clustering:** Implement K-Means clustering on Iris.csv dataset. Determine the number of clusters using the elbow method					
5	**Assignment on Random Forest Classifier:** Implement Random Forest Classifier model to predict the safety of the car.					
6	**Assignment on K-Means Clustering:** Build a Tic-Tac-Toe game using reinforcement learning in Python by using following tasks. a. Setting up the environment b. defining the Tic-Tac-Toe game c. Building the reinforcement learning model d. Training the model e. Testing the model					

Name & Signature of Course In-charge

Experiment No: 1

Feature Transformation PCA Algorithm

Name of the Student: ___________________________________

Class: **Roll No.:** **Batch:**

Date:

Mark: /10

Signature of the Course In-charge: _________________

Signature of the HOD: ____________________________

EXPERIMENT NO.1

Practical Title: Study PCA dimensionality reduction Technique

Aim: Apply PCA Algorithm on Wine Dataset and Distinguish between Red & White Wine.

Objective:

- To learn dimensionality reduction technique PCA and implement in Python.

Software Requirements:

- Anaconda with Python 3.7

Hardware Requirement:

- PIV, 2GB RAM, 500 GB HDD.

Learning Objectives:

To Learn PCA technique on given dataset.

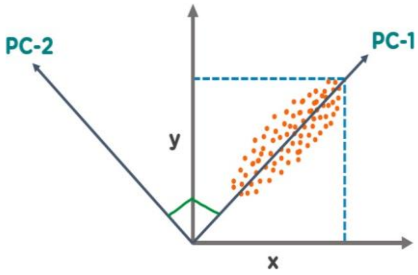
Outcome:

After completion of this assignment students are able to understand how is dimensionality reduction technique PCA work and how it will transform data from higher dimensions to lower dimensions and visualize using Matplotlib.

Theory:

Principal Component Analysis

The Principal Component Analysis is a popular unsupervised learning technique for reducing the dimensionality of data. It increases interpretability yet, at the same time, it minimizes information loss. It helps to find the most significant features in a dataset and makes the data easy for plotting in 2D and 3D. PCA helps in finding a sequence of linear combinations of variables.



In the above figure, we have several points plotted on a 2-D plane. There are two principal components. PC1 is the primary principal component that explains the maximum variance in the data. PC2 is another principal component that is orthogonal to PC1.

What is PCA

The Principal Components are a straight line that captures most of the variance of the data. They have a direction and magnitude. Principal components are orthogonal projections (perpendicular) of data onto lower- dimensional space. Now that you have understood the basics of PCA, let's look at the next topic on PCA in Machine Learning.

Dimensionality

The term "dimensionality" describes the quantity of features or variables used in the research. It can be difficult to visualize and interpret the relationships between variables when dealing with high-dimensional data, such as datasets with numerous variables. While reducing the number of variables in the dataset, dimensionality reduction methods like PCA are used to preserve the most crucial data. The original variables are converted into a new set of variables called principal components, which are linear combinations of the original variables, by PCA in order to accomplish this. The dataset's reduced dimensionality depends on how many principal components are used in the study. The objective of PCA is to select fewer principal components that account for the data's most important variation. PCA can help to streamline data analysis, enhance visualization, and make it simpler to spot trends and relationships between factors by reducing the dimensionality of the dataset.

The mathematical representation of dimensionality reduction in the context of PCA is as follows:
Given a dataset with n observations and p variables represented by the n x p data matrix X, the goal of PCA is to transform the original variables into a new set of k variables called principal components that capture the most significant variation in the data. The principal components are defined as linear combinations of the original variables given by:

PC_1 = a_11 * x_1 + a_12 * x_2 + ... + a_1p * x_p

PC_2 = a_21 * x_1 + a_22 * x_2 + ... + a_2p * x_p

...

PC_k = a_k1 * x_1 + a_k2 * x_2 + ... + a_kp * x_p where a_ij is the loading or weight of variable x_j on principal component PC_i, and x_j is the jth variable in the data matrix X. The principal components are ordered such that the first component PC_1 captures the most significant variation in the data, the second component PC_2 captures the second most significant variation, and so on. The number of principal components used in the analysis, k, determines the reduced dimensionality of the dataset.

Correlation

A statistical measure known as correlation expresses the direction and strength of the linear connection between two variables. The covariance matrix, a square matrix that displays the pairwise correlations between all pairs of variables in the dataset, is calculated in the setting of PCA using correlation. The covariance matrix's diagonal elements stand for each variable's variance, while the off-diagonal elements indicate the covariances between different pairs of variables. The strength and direction of the linear connection between two variables can be determined using the correlation coefficient, a standardized measure of correlation with a range of -1 to 1.

A correlation coefficient of 0 denotes no linear connection between the two variables, while correlation coefficients of 1 and -1 denote the perfect positive and negative correlations, respectively. The principal components in PCA are linear combinations of the initial variables that maximize the variance explained by the data. Principal components are calculated using the correlation matrix.

In the framework of PCA, correlation is mathematically represented as follows:

The correlation matrix C is a nxn symmetric matrix with the following components given a dataset with n variables (x_1, x_2,..., x_n):

$$C_{ij} = (sd(x_i) * sd(x_j)) / cov(x_i, x_j)$$

where sd(x_i) is the standard deviation of variable x_i and sd(x_j) is the standard deviation of variable x_j, and cov(x_i, x_j) is the correlation between variables x_i and x_j.

The correlation matrix C can also be written as follows in matrix notation:

$$C = X\text{^}T\ X / (n-1)\ (n-1)$$

Orthogonal

The term "orthogonality" alludes to the principal components' construction as being orthogonal to one another in the context of the PCA algorithm. This indicates that there is no redundant information among the main components and that they are not correlated with one another.

Orthogonality in PCA is mathematically expressed as follows: each principal component is built to maximize the variance explained by it while adhering to the requirement that it be orthogonal to all other principal components. The principal components are computed as linear combinations of the original variables. Thus, each principal component is guaranteed to capture a unique and non-redundant part of the variation in the data.

The orthogonality constraint is expressed as:

$$a\_i_1 * a\_j_1 + a\_i_2 * a\_j_2 + ... + a\_i_p * a\_j_p = 0$$

for all i and j such that i ≠ j. This means that the dot product between any two loading vectors for different principal components is zero, indicating that the principal components are orthogonal to each other.

Eigen Vectors

The main components of the data are calculated using the eigenvectors. The ways in which the data vary most are represented by the eigenvectors of the data's covariance matrix. The new coordinate system in which the data is represented is then defined using these coordinates.

The covariance matrix C in mathematics is represented by the letters v_1, v_2,..., v_p, and the associated eigenvalues are represented by _1, _2,..., _p. The eigenvectors are calculated in such a way that the equation shown below holds:

$$C\ v\_i = \lambda\_i\ v\_i$$

This means that the eigenvector v_i produces the associated eigenvalue λ_i as a scalar multiple of itself when multiplied by the covariance matrix C.

Covariance Matrix

The covariance matrix is crucial to the PCA algorithm's computation of the data's main components. The pairwise covariances between the factors in the data are measured by the covariance matrix, which is a p x p matrix. The correlation matrix C is defined as follows given a data matrix X of n observations of p variables: C = (1/n) * X^T X

where X^T represents X's transposition. The covariance's between the variables are represented by the off- diagonal elements of C, whereas the variances of the variables are represented by the diagonal elements of C.

Algorithm:

1. Import the Required Packages
2. Read Given Dataset
3. Import the Principal Component Analysis
4. Define input & output
5. Initialize the model
6. Fit the dataset
7. Draw Scatter Plot

Conclusion:

Thus we learn how to apply PCA algorithm on Wine dataset and visualize the classes by plotting on 2D graph.

Viva Questions:

1. What is Machine Learning?
2. What is PCA Algorithm?
3. Difference between PCA and LDA?
4. What are Different Types of ML?

Coding Efficiency	Viva	Timely Completion	Total	Dated Sign of Course In-charge
5	3	2	10	

Experiment No: 2

Assignment based on Linear Regression

Name of the Student: ______________________________

Class: **Roll No.:** **Batch:**

Date:

Mark: /10

Signature of the Course In-charge: __________________

Signature of the HOD: _____________________________

EXPERIMENT NO.2

Practical Title: Assignment based on Linear Regression

Aim: Predict the price of the Uber ride from a given pickup point to the agreed drop-off location. Perform following tasks:

1. Pre-process the dataset.
2. Identify outliers.
3. Check the correlation.
4. Implement linear regression and random forest regression models.
5. Evaluate the models and compare their respective scores like R2, RMSE, etc.

Prerequisite:

1. Basic knowledge of Python
2. Concept of preprocessing data
3. Basic knowledge of Data Science and Big Data Analytics.

Software Requirements:

- Anaconda with Python 3.7

Hardware Requirement:

- PIV, 2GB RAM, 500 GB HDD

Learning Objectives:

Students should be able to preprocess dataset and identify outliers, to check correlation and implement linear regression and random forest regression models. Evaluate them with respective scores like R2, RMSE etc.

Theory:

Data Preprocessing:

Data Preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data preprocessing task.

Why do we need Data Preprocessing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

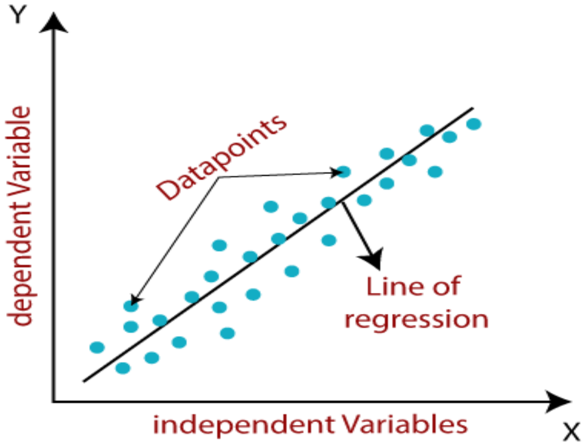
- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

Linear Regression:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:

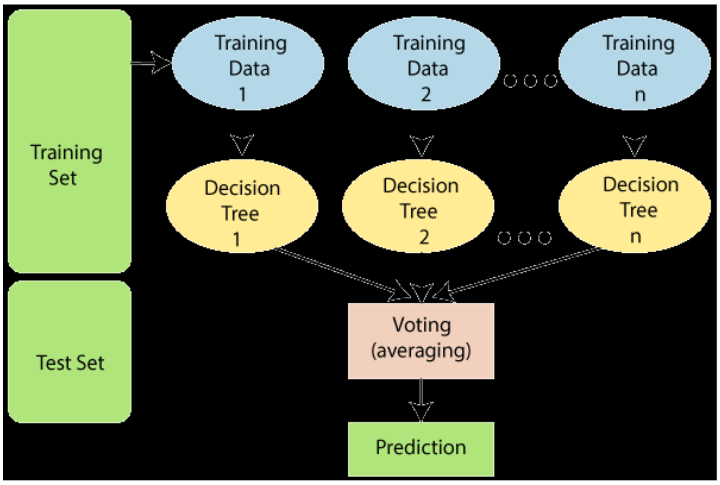


Random Forest Regression Models:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

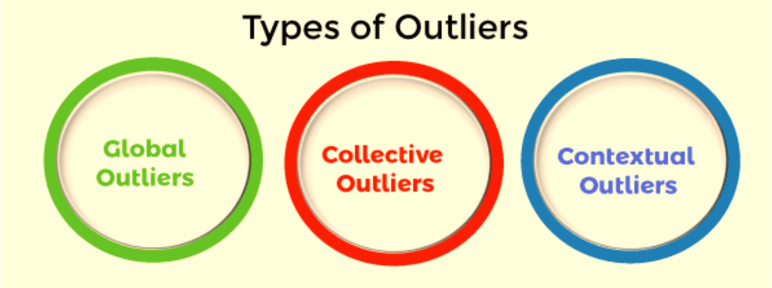
As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. "Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



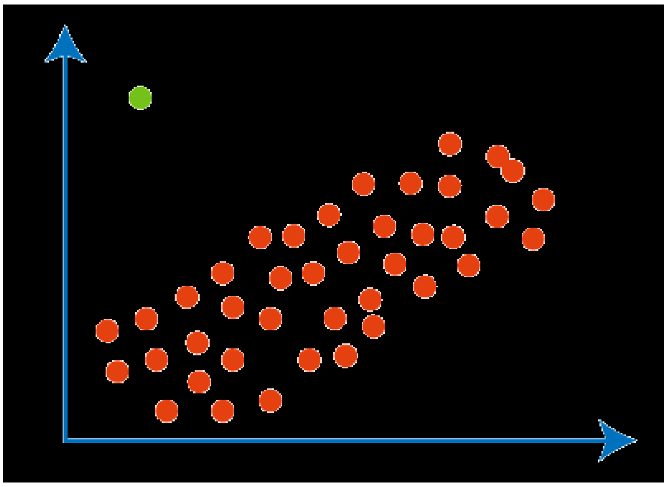
Outliers:

As the name suggests, "outliers" refer to the data points that exist outside of what is to be expected. The major thing about the outliers is what you do with them. If you are going to analyze any task to analyze data sets, you will always have some assumptions based on how this data is generated. If you find some data points that are likely to contain some form of error, then these are definitely outliers, and depending on the context, you want to overcome those errors. The data mining process involves the analysis and prediction of data that the data holds. In 1969, Grubbs introduced the first definition of outliers.



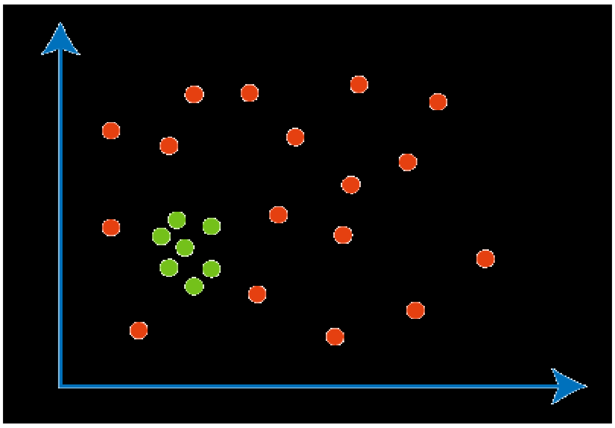
Global Outliers

Global outliers are also called point outliers. Global outliers are taken as the simplest form of outliers. When data points deviate from all the rest of the data points in a given data set, it is known as the global outlier. In most cases, all the outlier detection procedures are targeted to determine the global outliers. The green data point is the global outlier.



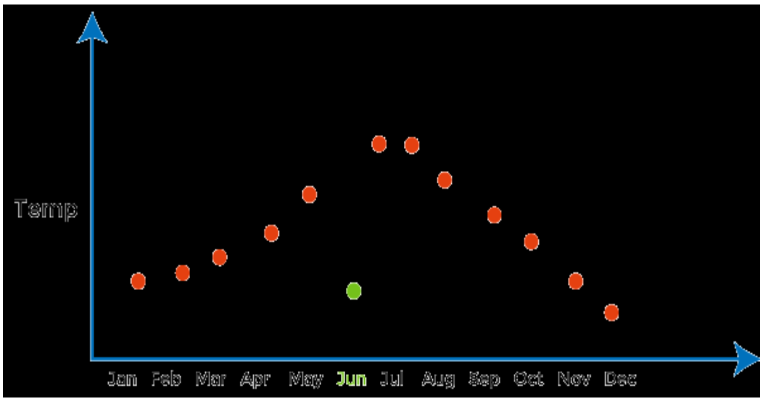
Collective Outliers

In a given set of data, when a group of data points deviates from the rest of the data set is called collective outliers. Here, the particular set of data objects may not be outliers, but when you consider the data objects as a whole, they may behave as outliers. To identify the types of different outliers, you need to go through background information about the relationship between the behavior of outliers shown by different data objects. For example, in an Intrusion Detection System, the DOS package from one system to another is taken as normal behavior. Therefore, if this happens with the various computer simultaneously, it is considered abnormal behavior, and as a whole, they are called collective outliers. The green data points as a whole represents the collective outlier.



Contextual Outliers

As the name suggests, "Contextual" means this outlier introduced within a context. For example, in the speech recognition technique, the single background noise. Contextual outliers are also known as Conditional outliers. These types of outliers happen if a data object deviates from the other data points because of any specific condition in a given data set. As we know, there are two types of attributes of objects of data: contextual attributes and behavioral attributes. Contextual outlier analysis enables the users to examine outliers in different contexts and conditions, which can be useful in various applications. For example, A temperature reading of 45 degrees Celsius may behave as an outlier in a rainy season. Still, it will behave like a normal data point in the context of a summer season. In the given diagram, a green dot representing the low-temperature value in June is a contextual outlier since the same value in December is not an outlier.



Mean Squared Error:

The **Mean Squared Error (MSE)** or **Mean Squared Deviation (MSD)**of an estimator measures the average of error squares i.e. the average squared difference between the estimated values and true value. It is a risk function, corresponding to the expected value of the squared error loss. It is always non – negative and values close to zero are better. The MSE is the second moment of the error (about the origin) and thus incorporates both the variance of the estimator and its bias.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE	= mean squared error
n	= number of data points
Y_i	= observed values
$\hat{Y}_i$	= predicted values

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

RMSD	= root-mean-square deviation
i	= variable i
N	= number of non-missing data points
x_i	= actual observations time series
$\hat{x}_i$	= estimated time series

Algorithm:

1. Import the Required Packages
2. Read Given Dataset
3. Import the Linear Regression
4. Data Preprocessing
5. Define input & output
6. Initialize the model
7. Fit the dataset
8. Evaluate the model

Conclusion:

Thus we learn how to implement Linear Regression model with Ridge & Lasso Regression models on uber dataset.

Viva Questions:

1. What is Linear Regression?
2. Explain Ridge & Lasso Regression?
3. What are the different types of Outliers?
4. What is the mean square error in ML?
5. What is R2, RMSE?

Coding Efficiency	Viva	Timely Completion	Total	Dated Sign of Course In-charge
5	3	2	10	

Experiment No: 3

Assignment based on SVM Classification

Name of the Student: ______________________________

Class: **Roll No.:** **Batch:**

Date:

Mark: /10

Signature of the Course In-charge: __________________

Signature of the HOD: _____________________________

EXPERIMENT NO.3

Practical Title: Assignment based on SVM Classification

Aim: Implementation of Support Vector Machines (SVM) for classifying images of handwritten digits into their respective numerical classes (0 to 9).

Prerequisite:

- Basic of Python, Data Mining Algorithm, Concept of KNN Classification

Software Requirements:

- Anaconda with Python 3.7

Hardware Requirement:

- PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089Model.

Learning Objectives:

- Understand the fundamental theory behind Support Vector Machines (SVM).
- Implement an SVM algorithm using a standard machine learning library.
- Apply SVM to a real-world dataset (e.g., MNIST) for classifying handwritten digits.
- Evaluate the performance of the SVM classifier.
- Interpret the results and understand the practical applications and limitations of SVM.

Objectives:

- To learn SVM for classifying images of handwritten digits.

Outcome:

- Confusion Matrix: Provides a summary of the prediction results on the test dataset.
- Classification Report: Includes precision, recall, f1-score, and support for each class.

Theory:

Classification Analysis:

Definition: This analysis is a data mining technique used to determine the structure and categories within a given dataset. Classification analysis is commonly used in machine learning, text analytics, and statistical modelling. Above all, it can help identify patterns or groupings between individual observations, enabling researchers to understand their datasets better and make more accurate predictions.

Classification analysis is used to group or classify objects according to shared characteristics. Moreover, this analysis can be used in many applications, from segmenting customers for marketing campaigns to forecasting stock market trends.

Classification Analysis Example:

- **Classifying images:**

One example of a classification analysis is the use of supervised learning algorithms to classify images. In this case, the algorithm is provided with an image dataset (the training set) that contains labelled images. The algorithm uses labels to learn how to distinguish between different types of objects in the picture. Once trained, it can then be used to classify new images as belonging to one category or another.

- **Customer Segmentation:**

Another example of classification analysis would be customer segmentation for marketing campaigns. Classification algorithms group customers into segments based on their characteristics and behaviors. This helps marketers target specific groups with tailored content, offers, and promotions that are more likely to appeal to them.

- **Stock Market Prediction:**

Finally, classification analysis can also be used for stock market prediction. Classification algorithms can identify patterns between past stock prices and other economic indicators, such as interest rates or unemployment figures. By understanding these correlations, analysts can better predict future market trends and make more informed investment decisions.

Support Vector Machine:

Support Vector Machine is one of the most popular Supervised Learning Algorithms, which is used for Classification as well as Regression Problem. However, primarily, it is used for Classification Problem in Machine Learning.

Key Concepts of SVM

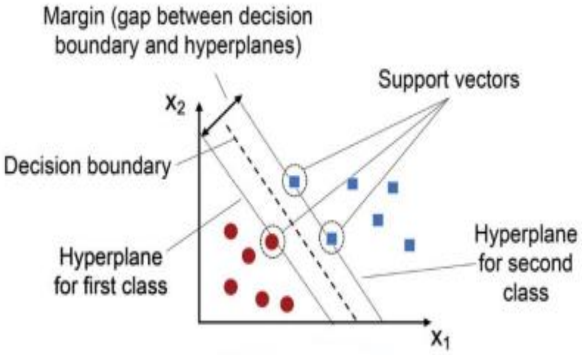
- **Support Vector** –Data point that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.
- **Hyperplane-** It is a decision plane or space which is divided between a set of objects having different classes.
- **Margin –** It may be defined as the gap between two line on the closet data point of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margined and small margin is considered as a bad margin.

SVM is a supervised learning algorithm used for classification and regression tasks. It works by finding the hyperplane that best separates the data into different classes.

1. **Hyperplanes and Margins:**
   - A hyperplane is a decision boundary that separates data points of different classes.
   - The goal is to find the hyperplane that maximizes the margin, which is the distance between the hyperplane and the nearest data points from each class (support vectors).

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

2. **Kernel Trick:**
   - In cases where data is not linearly separable, kernels are used to transform the data into a higher-dimensional space where a hyperplane can separate the classes.
   - Common kernels include linear, polynomial, and radial basis function (RBF).
3. **Regularization:**
   - SVM includes a regularization parameter (C) that controls the trade-off between maximizing the margin and minimizing classification errors.
   - A larger C value prioritizes classification accuracy over margin width, while a smaller C value emphasizes a wider margin.


Margin (gap between decision boundary and hyperplanes)
Support vectors
X2
Decision boundary
Hyperplane for first class
Hyperplane for second class
X1

Dataset:

MNIST Dataset: A widely used dataset for handwritten digit classification. It consists of 70,000 grayscale images of handwritten digits (0-9), with each image being 28x28 pixels.

Conclusion:

Implementing SVM for handwritten digit classification involves understanding the theoretical aspects of SVM, preparing and preprocessing the data, training the SVM model, and evaluating its performance

Viva Questions:

1. What are Support Vector Machines (SVMs)?
2. What are Support Vectors in SVMs?
3. What is the basic principle of a Support Vector Machine?
4. What are hard margin and soft Margin SVMs?
5. What do you mean by Hinge loss?

Coding Efficiency	Viva	Timely Completion	Total	Dated Sign of Course In-charge
5	3	2	10	

Experiment No: 4

Assignment on K-Means Clustering:

Name of the Student: ______________________________

Class: **Roll No.:** **Batch:**

Date:

Mark: /10

Signature of the Course In-charge: __________________

Signature of the HOD: _____________________________

EXPERIMENT NO.4

Practical Title: Implementation K-Means clustering

Aim: Implementation K-Means clustering on iris.csv dataset. Determine the number of clusters using the elbow method.

Prerequisite:

- Basic of Python, Data Mining Algorithm, Concept of K-mean Clustering

Software Requirements:

- Anaconda with Python 3.7

Hardware Requirement:

- PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089 Model.

Learning Objectives:

- To learn the basics of clustering.

Objectives:

1. Understand the Iris Dataset.
2. Implement K-Means Clustering.
3. Determine the Optimal Number of Clusters Using the Elbow Method.
4. Analyze and Interpret the Results.

Outcome:

- Successful Data Loading and Exploration.
- K-Means Clustering Implementation.
- Visualization of Clustering Results

Theory:

A Hospital Care chain wants to open a series of Emergency-Care wards within a region. We assume that the hospital knows the location of all the maximum accident-prone areas in the region. They have to decide the number of the Emergency Units to be opened and the location of these Emergency Units, so that all the accident-prone areas are covered in the vicinity of these Emergency Units.

What is Clustering?

Clustering is a type of **unsupervised machine learning** technique used to group or categorize a set of objects (data points) into clusters or groups based on their similarity. The primary objective is to ensure that objects within the same cluster are more similar to each other than to those in other clusters. Clustering helps identify patterns, relationships, and structures within data when there are no predefined labels or categories.

Application of Clustering:

Clustering is used in almost all the fields. You can infer some ideas from Example 1 to come up with lot of clustering applications that you would have come across.

Listed here are few more applications, which would add to what you have learnt.

- ✓ Clustering helps marketers improve their customer base and work on the target areas. It helps group people (according to different criteria's such as willingness, purchasing power etc.) based on their similarity in many ways related to the product under consideration.
- ✓ Clustering helps in identification of groups of houses on the basis of their value, type and geographical locations.
- ✓ Clustering is used to study earth-quake. Based on the areas hit by an earthquake in a region, clustering can help analyse the next probable location where earthquake can occur.

What is K-means Clustering?

K-Means Clustering Algorithm K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm. The k-means clustering algorithm mainly performs two tasks:

● Determines the best value for K center points or centroids by an iterative process.

● Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

K-means Clustering Method:

If k is given, the K-means algorithm can be executed in the following steps:

- Partition of objects into k non-empty subsets
- Identifying the cluster centroids (mean point) of the current partition.
- Assigning each point to a specific cluster
- Compute the distances from each point and allot points to the cluster where the distance from the centroid is minimum.
- After re-allotting the points, find the centroid of the new cluster formed.

Elbow Method

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. WCSS stands for Within Cluster Sum of Squares, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

WCSS= $\sum_{P_i \text{ in Cluster1}} \text{distance}(P_i\ C_1)^2 + \sum_{P_i \text{ in Cluster2}} \text{distance}(P_i\ C_2)^2 + \sum_{P_i \text{ in CLuster3}} \text{distance}(P_i\ C_3)^2$

In the above formula of WCSS,

$\sum_{P_i \text{ in Cluster1}} \text{distance}\ (P_i\ C_1)^2$: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

- It executes the K-means clustering on a given dataset for different K values (Ranges from 1-10).
- For each value of K, calculates the WCSS value.
- Plots a curve between calculated WCSS values and the number of clusters K.

Conclusion:

Successfully implement K-Means clustering using the elbow method.

Viva Questions:

1. Why does k-means clustering algorithm use only Euclidean distance metric?
2. How to decide on the correct number of clusters?
3. What are the Different Application of K Mean Clustering?
4. What is K-medoids?
5. What is k-medians clustering?
6. Explain K-Means Clustering?
7. Explain Hierarchical Clustering?
8. Explain Elbow Method?

Coding Efficiency	Viva	Timely Completion	Total	Dated Sign of Course In-charge
5	3	2	10	

Experiment No: 5

Implement Random Forest Classifier

Name of the Student: ______________________________

Class: **Roll No.:** **Batch:**

Date:	Mark: /10

Signature of the Course In-charge: __________________

Signature of the HOD: _____________________________

EXPERIMENT NO. 5

Practical Title: Implement Random Forest Classifier

Aim: Implement Random Forest Classifier model to predict the safety of the car.

Software Requirements:

- Anaconda with Python 3.7

Hardware Requirement:

- PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089Model.

Learning Objectives:

- To learn the basics concept of Random Forest Classifier Algorithm.

Objectives:

- Learn the working of ensemble learning techniques and how Random Forest combines multiple decision trees to improve prediction accuracy.
- Explore how Random Forest handles both classification and regression tasks but focus on classification for predicting car safety.
- Use techniques such as grid search or random search to improve model performance by tuning these parameters.

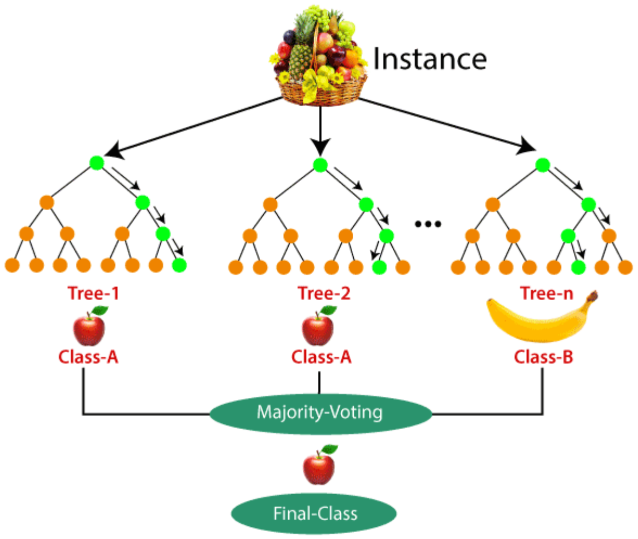
Outcome:

- Improved Understanding of Machine Learning Algorithms
- Learn how Random Forest can generalize well on different datasets, making the model useful in various car safety prediction applications.
- Obtain a more robust and reliable classifier compared to a single decision tree model, thanks to the ensemble nature of Random Forest, which helps in reducing overfitting and improving generalization.

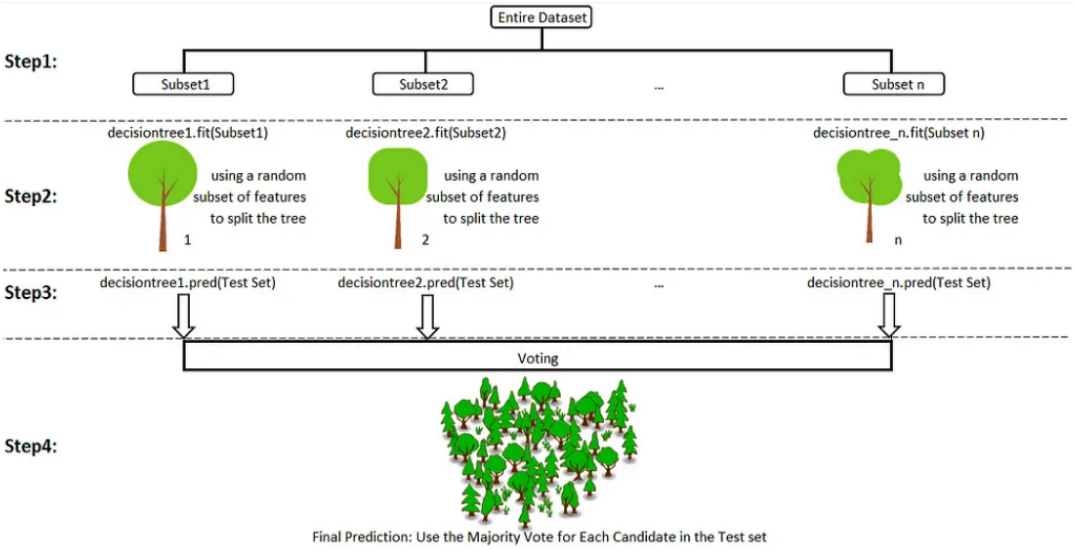
Theory:

Random Forest Classifier

Random forest is supervised learning algorithm that is used widely in classification an regression problem. It builds decision trees on different samples and takes their majority vote for classification and average in case of regressions. Random forest and bagging are "bagging" algorithm that aim to reduces the complexity of models that overfit the training data. Random forest tries to give more preferences to hyper parameters to optimize the model. It is handle binary, continuous and categorical data.
Random forest is both a supervised learning algorithm and an ensemble algorithm.



Random Forest is an ensemble model using bagging as the ensemble method and decision tree as the individual model.


Entire Dataset
Step1:
Subset1
Subset2
...
Subset n
Step2:
decisiontree1.fit(Subset1)
decisiontree2.fit(Subset2)
decisiontree_n.fit(Subset n)
using a random subset of features to split the tree
1
using a random subset of features to split the tree
2
using a random subset of features to split the tree
n
Step3:
decisiontree1.pred(Test Set)
decisiontree2.pred(Test Set)
...
decisiontree_n.pred(Test Set)
Voting
Step4:
Final Prediction: Use the Majority Vote for Each Candidate in the Test set

Random forests are a popular supervised machine learning algorithm.

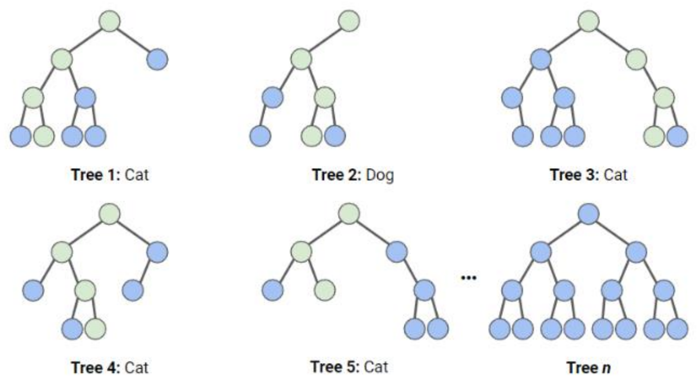
- Random forests are for supervised machine learning, where there is a labeled target variable.
- Random forests can be used for solving regression (numeric target variable) and classification (categorical target variable) problems.
- Random forests are an ensemble method, meaning they combine predictions from other models.
- Each of the smaller models in the random forest ensemble is a decision tree.

How Random Forest Classification Works

Imagine you have a complex problem to solve, and you gather a group of experts from different fields to provide their input. Each expert provides their opinion based on their expertise and experience. Then, the experts would vote to arrive at a final decision.

In a random forest classification, multiple decision trees are created using different random subsets of the data and features. Each decision tree is like an expert, providing its opinion on how to classify the data. Predictions are made by calculating the prediction for each decision tree, then taking the most popular result. (For regression, predictions use an averaging technique instead.)

In the diagram below, we have a random forest with n decision trees, and we've shown the first 5, along with their predictions (either "Dog" or "Cat"). Each tree is exposed to a different number of features and a different sample of the original dataset, and as such, every tree can be different. Each tree makes a prediction. Looking at the first 5 trees, we can see that 4/5 predicted the sample was a Cat. The green circles indicate a hypothetical path the tree took to reach its decision. The random forest would count the number of predictions from decision trees for Cat and for Dog, and choose the most popular prediction.



Application:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm

Conclusion:

Implement Random Forest Classifier model to predict the safety of the car using Ensemble Learning.

Viva Questions:

1. What is a Random Forest Classifier?
2. How does a Random Forest work?
3. What is the difference between a Decision Tree and a Random Forest?
4. Why is Random Forest considered an ensemble method?
5. What are the advantages of using Random Forest over a single Decision Tree?
6. Can Random Forest be used for both classification and regression tasks?
7. Explain the concept of bagging in the context of Random Forest.
8. What is the role of randomization in Random Forest?

Coding Efficiency	Viva	Timely Completion	Total	Dated Sign of Course In-charge
5	3	2	10	

Experiment No: 6

Build a Tic-Tac-Toe game using reinforcement learning

Name of the Student: ______________________________

Class: **Roll No.:** **Batch:**

Date:

Mark: /10

Signature of the Course In-charge: __________________

Signature of the HOD: _____________________________

EXPERIMENT NO. 6

Practical Title: Build a Tic-Tac-Toe game using reinforcement learning

Aim: Build a Tic-Tac-Toe game using reinforcement learning in Python by using following tasks

a. Setting up the environment

b. Defining the Tic-Tac-Toe game

c. Building the reinforcement learning model

d. Training the model e. Testing the model

Prerequisite:

- Basic of Python, Data Mining Algorithm, Reinforcement Learning.

Software Requirements:

- Anaconda with Python 3.7

Hardware Requirement:

- PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089Model.

Learning Objectives:

- To learn the basics concept of reinforcement learning.

Objectives:

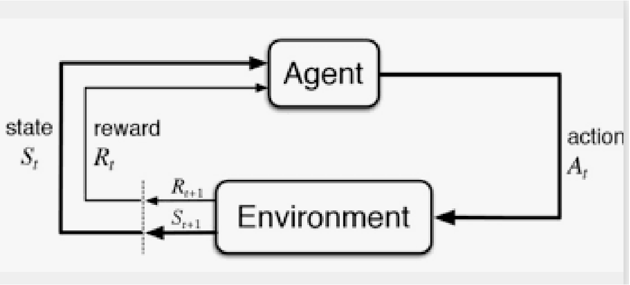
1. To implement a Tic-Tac-Toe game with reinforcement learning (RL).
2. To implement a Tic-Tac-Toe game using reinforcement learning (RL) to train an agent that learns optimal strategies through interaction with the game environment.

Outcome:

- To understand RL and Q-learning concepts, gain practical experience in game development and agent training, and be able to implement and evaluate an AI strategy in Tic-Tac-Toe.

Theory:

Reinforcement Learning (RL) is a type of machine learning where an agent learns to make decisions by taking actions in an environment to maximize cumulative reward. The agent learns through trial and error, receiving feedback from its actions in the form of rewards or penalties.



Reinforcement learning involves an agent making decisions in an environment to maximize cumulative rewards. The Q-learning algorithm updates a Q-table that stores the value of state-action pairs, guiding the agent towards optimal moves. By playing multiple games, the agent improves its strategy through trial and error, eventually learning to play optimally.

Key concepts in RL:

- **Agent**: The player or decision-maker.
- **Environment**: The world the agent interacts with (in this case, the Tic-Tac-Toe board).
- **State**: A representation of the environment at a particular time.
- **Action**: A move made by the agent.
- **Reward**: Feedback from the environment after an action.
- **Policy**: A strategy used by the agent to decide on actions.
- **Value Function**: A function that estimates the expected reward for a given state or state-action pair.

Q-Learning

Q-learning is a machine learning approach that enables a model to iteratively learn and improve over time by taking the correct action. Q-learning is a type of reinforcement learning. This approach to reinforcement learning takes the opposite approach. The agent receives no policy, meaning its exploration of its environment is more self-directed.

Q-Value Update Formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

where:

- Q(s,a) is the current Q-value for state s and action a.
- α\alphaα is the learning rate, controlling how much new information overrides the old.
- r is the immediate reward received after taking action a in state s.
- γ\gamma γ is the discount factor, representing the importance of future rewards.
- $\max_a$ Q(s′, a′) is the maximum Q-value for the next state s' over all possible actions a'.

Q-learning Algorithm Steps:

1. **Initialize Q-Table:** Create a Q-table where each entry represents the Q-value for a state-action pair. Initially, all values are set to zero.
2. **Choose Action:** Use an exploration-exploitation strategy to select actions. The exploration strategy involves trying new actions randomly, while exploitation focuses on choosing actions that maximize the known Q-value.
3. **Take Action:** Perform the action in the environment, observe the resulting state, and receive a reward.
4. **Update Q-value:** Adjust the Q-value for the state-action pair based on the reward received and the maximum future reward predicted from the new state.

Deep Q-Networks:

These algorithms utilize neural networks in addition to reinforcement learning techniques. They utilize the self-directed environment exploration of reinforcement learning. Future actions are based on a random sample of past beneficial actions learned by the neural network.

Conclusion:

Implement Tic-Tac-Toe game where an agent learns to play using reinforcement learning.

Viva Questions:

1. What is reinforcement learning, and how does it differ from supervised and unsupervised learning?
2. Explain the concept of the Q-table in Q-learning and how it is used to store and update state-action values.
3. can you explain the key components of the Q-learning algorithm and how they are applied to train an agent in the Tic-Tac-Toe game?
4. What is the primary objective of using reinforcement learning for a Tic-Tac-Toe game, and how does it differ from traditional rule-based approaches?

Coding Efficiency	**Viva**	**Timely Completion**	**Total**	**Dated Sign of Course In-charge**
5	**3**	**2**	**10**	